Extending the scalars of minimizations

G. DUCHAMP[1] É. LAUGEROTTE[2]
Laboratoire d'**I**nformatique **F**ondamentale et **A**ppliquée de **R**ouen
Faculté des Sciences et des Techniques
76821 Mont Saint Aignan CEDEX France

J-G. LUQUE[3]
Institut **G**aspard **M**onge
Université de Marne la Vallée
77454 Marne la Vallée France

1 Introduction

In the classical theory of formal languages, finite state automata allow to recognize the words of a rational subset of Σ^* where Σ is a set of symbols (or the alphabet). Now, given a semiring $(\mathbb{K}, +, .)$, one can construct $\mathbb{K}$-subsets of Σ^* in the sense of Eilenberg [5], that are alternatively called noncommutative formal power series [2, 13] for which a framework very similar to language theory has been constructed (see [11, 12] and [2]). This extension has applications in many domains. Let us cite, for example, enumeration (non-commutative as used by instance for alignment of genomic sequences), image processing [3], task-ressource problems [8] and real-time applications where multiplicities are used to prove the modularity of the validation method by means of the Hadamard product of two integer valued automata (see the contribution by Geniet and Dubernard [9]).

Particular noncommutative formal power series, which are called rational series, are the behaviour of a family of weighted automata (or $\mathbb{K}$-automata). In order to get an efficient encoding, it may be interesting to point out one of them with the smallest number of states. Minimization processes of $\mathbb{K}$-automata already exist for $\mathbb{K}$ being:
a) a field [2],
b) a noncommutative field [7],
c) a PID [6].
When $\mathbb{K}$ is the bolean semiring, such a minimization process (with isomor-

[1]Gerard.Duchamp@univ-rouen.fr
[2]Eric.Laugerotte@univ-rouen.fr
[3]Jean-Gabriel.Luque@univ-mlv.fr
Partially supported by the Scientific Research Program of MENRT

phisms of minimal objects) is known within the category of deterministic automata.

Minimal automata have been proved to be isomorphic in cases **(a)** and **(b)** (see respectively [2] and [7]). The case **(c)** is mentioned in [6]. But the proof given in [2] is not constructive. In fact, it lays on the existence of a basis for a submodule of $\mathbb{K}^n$. Here we give an independent algorithm which reproves this fact and an example of a pair of nonisomorphic minimal automata. Moreover, we examine the possibility of extending **(c)**. To this end, we provide an *Effective Minimization Process* (or *EMP*) which can be used for more general sets of coefficients.

The structure of the contribution is the following. After this introduction, we give in details the EMP and, in Section 3, we discuss the termination of the EMP in a frame which extends **(c)**.

2 Computing a prefix subset

Let $\mathbb{K}$ be an integral domain (a ring without zero divisor) and Σ a finite alphabet. A $\mathbb{K}$-automaton $\mathcal{A}$ is usually identified by a linear representation (λ, μ, γ). We examine here a process which allows us to find a prefix subset X such that $\lambda\mu(X)$ generates $\lambda\mu(\mathbb{K}\langle A\rangle)$. We apply Algorithm **prefix** (which calls Algorithm **generator**) to a $\mathbb{K}$-automaton $\mathcal{A}$.

Algorithm prefix
input : the linear representation (λ, μ, γ).
output : a pair (X, Z) where X is a prefix code and $Z \subset X$.

1. $(X_0, Y_0, Z_0) := (\emptyset, \{1\}, \emptyset)$
2. if $Y_i \neq \emptyset$
   (a) choose $y \in Y_i$ of minimal length
   (b) $(X_{i+1}, Y_{i+1}, Z_{i+1}) :=$ **generator**$((\lambda, \mu, \gamma), y, (X_i, Y_i, Z_i))$
   (c) go to (2)
3. return (X, Z)

Algorithm generator
input : the linear representation (λ, μ, γ),
the word y,
the triplet (X, Y, Z).
output : the triplet (X, Y, Z).

1. $n := |X|$
2. if it does not exist $\alpha \in \mathbb{K}$ such that $\alpha\lambda\mu(y) = \alpha_1\lambda\mu(x_1)+\cdots+\alpha_n\lambda\mu(x_n)$ with $\alpha_i \in K$ and $x_i \in X$ $(1 \leq i \leq n)$ then

   $$(X, Y, Z) := (X \cup \{y\}, Y \cup yA \setminus \{y\}, Z)$$

3. else if it exists such a α which divides α_i $(1 \leq i \leq n)$

   $$(X, Y, Z) := (X, Y \setminus \{y\}, Z)$$

4. else

   $$(X, Y, Z) := (X \cup \{y\}, Y \cup yA \setminus \{y\}, Z \cup \{y\})$$

5. return (X, Y, Z)

As a computation process [10], Algorithm **prefix** is well-defined if we can compute α and the α_i's in Algorithm **generator**. Let us denote $\mathbb{F}$ the field of fractions of $\mathbb{K}$.

Proposition 1 *When the computation process terminates,*

1. *the family $\lambda\mu(X)$ generates $\lambda\mu(\mathbb{K}\langle A\rangle)$.*
2. *the family $\lambda\mu(X - Z)$ is linearly independent for $\mathbb{F}$.*

We prove 1 as in [2] using the decomposition $\Sigma^* = C^*X$ with $C = (X\Sigma \cup \epsilon) \setminus X$ the prefix code induced by X, and using the linearity of μ. Now, for 2, the only way to make the set $X - Z$ increasing is to come through Step 2 of Algorithm **generator** where we add to X an item y such that $\lambda\mu(y)$ is linearly independant of $\lambda\mu(X - Z)$.

The family $\lambda\mu(X - Z)$ does not generate necessarily $\lambda\mu(\mathbb{K}\langle A\rangle)$ but we could expect that it exists a basis of $\lambda\mu(\mathbb{K}\langle A\rangle)$ of rank $|X - Z|$. This occurs only when the relation $\alpha\lambda\mu(y) = \sum_{x\in X} \alpha_x\lambda\mu(x)$ implies that the rank of $\mathrm{Span}(\lambda\mu(X \cup \{y\}))$ is $|X|$. Or again, this is equivalent to respect the following condition: for each n, $m \in \mathbb{N}^+$, if $V = \{v_i\}_{i\in[1,m]} \subseteq \mathbb{K}^n$ is linearly independent and $\alpha u = \sum_i \alpha_i v_i$ (with $\alpha \in \mathbb{K} - \{0\}$) then the rank of $\mathrm{Span}(V \cup \{u\})$ is m. Then, setting $m = 1$ and $n = 1$, we find that such a ring $\mathbb{K}$ verifies the Bézout condition. We will say that $\mathbb{K}$ is a Bézout ring. Conversely, suppose that $\mathbb{K}$ is a Bézout ring. Using a Gauss method, we find

the property. More precisely, let $\binom{a}{b} \in \mathbb{K}^2$. If $b = 0$ the triangularization is clear. If $a = 0$ then

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \binom{0}{b} = \binom{b}{0}.$$

Otherwise, as $\mathbb{K}$ is a Bézout ring, it exists $\alpha, \beta \in \mathbb{K}$ such that

$$\alpha a + \beta b = \gcd(a, b) = d.$$

Then, if the matrix G is defined by

$$G = \begin{pmatrix} \alpha & \beta \\ -\frac{b}{d} & \frac{a}{d} \end{pmatrix}$$

one has

$$G \binom{a}{b} = \binom{d}{0}.$$

But, as $\mathbb{K}$ is an integral domain, the matrix G is unimodular. We can apply this process to triangularize matrices in $\mathbb{K}^{n\times n}$. In the sequel, we will only consider an integral Bézout domain $\mathbb{K}$.

3 Minimization in integral Bézout domains

We use the prefix code computed in Algorithm **prefix** in order to construct a left reduced of a linear representation (λ, μ, γ). The main step of our algorithm is to choose a basis of $\lambda\mu\left(\mathbb{K}\langle A\rangle\right)$ using the previous Gauss process. In fact, we consider Algorithm **triang** taking a matrix M as input and returning a pair (G, T) where T is a stair matrix and G a Gauss matrix such that

$$\binom{T}{\overline{0^n}} = \begin{pmatrix} G & 0 \\ 0 & Id_n \end{pmatrix} M$$

with n maximal.

Algorithm left_reduction
input : a linear representation (λ, μ, γ).
output : a left reduced linear representation $(\lambda_r, \mu_r, \gamma_r)$.

1. $(X, Z) := \mathbf{prefix}((\lambda, \mu, \gamma))$
2. $(I, T) = ((1), (\lambda))$
3. if $X \neq \emptyset$ then
   (a) choose $x \in X$ of minimal length
   (b) $X := X \setminus \{x\}$
   (c) $(G, T) := \mathbf{triang}\left(\left(\frac{T}{\lambda\mu(x)}\right)\right)$
   (d) if $x \in Z$ then $I := IG^{-1}$ else $I := (I|0)G^{-1}$
   (e) go to (3)
4. for each $a \in A$, compute $\mu_r(a)$ such that $T\mu(a) = \mu_r(a)T$.
5. return $(I, \mu_r, T\gamma)$

In Step 3(d), the "if" part occurs when the rank of the matrix $\left(\left(\frac{T}{\lambda\mu(x)}\right)\right)$ with coefficients in $\mathbb{Z}$ is equal to the rank of T. In the "else" part, we add an item to the family, and then one line to T and one column to I.

Proposition 2 *Let $\mathcal{A} = (\lambda, \mu, \gamma)$ be a linear representation. When the computational method terminates, Algorithm* **left_reduction** *gives a left reduced $\mathbb{K}$-automaton of $\mathcal{A}$.*

We can observe that $\lambda\mu(w)\gamma = \lambda_r T\mu(w)\gamma = \lambda_r\mu_r(w)T\gamma = \lambda_r\mu_r(w)\gamma_r$. Furthermore, the construction implies that the linear representation $\lambda_r\mu_r(\mathbb{F}\langle X\rangle)$ lies in $\mathbb{F}^{1\times|X-Z|}$. Moreover, we can compute a right reduced automaton using the previous algorithm with the linear representation $(\gamma^t, \mu^t, \lambda^t)$ as input. Realizing a left reduction and a right reduction gives a minimal $\mathbb{K}$-automaton. However, here, two minimal linear representations are not necessarily isomorphic. As shown the following example:

1. $\mathcal{A}_1 = \left((1 \quad 0), \begin{pmatrix} 0 & x \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right)$
2. $\mathcal{A}_2 = \left((x \quad 0), \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right)$

with $x \in \mathbb{Z} - \{0, 1, -1\}$. These $\mathbb{K}$-automata are different minimal linear representations with a same behaviour, but they are not isomorphic. Let T be a matrix such that

$$\mathcal{A}_1 = \left((x \quad 0)T^{-1}, T \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} T^{-1}, T \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right).$$

This relation implies necessarily that

$$(T^{-1})_{2,1} = \frac{1}{x} \notin \mathbb{Z}.$$

Note Let $\mathbb{F}$ be a field, then $\mathbb{K} = \mathbb{F}((X^\alpha)_{\alpha \in \mathbb{Q}_+ \setminus \{0\}})$ (polynomials with fractional powers) provides an example of integral Bézout domain where Algorithm **left_reduction** terminates, and which is not a principal integral domain.

References

[1] ANDARY P., CARON P., CHAMPARNAUD J-M., DUCHAMP G., FLOURET M. and LAUGEROTTE É., *SEA: a Symbolic Environement for Automata*, Proceedings of WIA'99 (Postdam 1999).

[2] BERSTEL J. and REUTENAUER C., *Rational series and their languages* (Springer-Verlag, 1988).

[3] CULIK II K. and KARI J., *Finite state transformations of images*, Proceedings of ICALP 95, Lecture Notes in Comput. Sci. **944** 51-62 (1995).

[4] DUCHAMP G., FLOURET M., LAUGEROTTE É., LUQUE J-G., *Direct and dual laws for automata with multiplicities*, Theoret. Comput. Sci. , to appear.

[5] EILENBERG S., *Automata, languages and machines, Vol. A* (Academic Press, 1974).

[6] FLIESS M., *Matrices de Hankel*, J. Math. Pures et Appl. **53** 197-224 (1974).

[7] FLOURET M. and LAUGEROTTE É., *Noncommutative minimization algorithms*, Inform. Process. Lett. **64** 123-126 (1997).

[8] GAUBERT S. and MAIRESSE J., *Task ressource models and (max,+) automata* in *Idempotency*, Publi. of the Isaac Newton Institute 133-144 (Cambridge Univ. Press, 1998).

[9] GENIET D. and DUBERNARD J-P, *Association de langages rationnels et de fonctions génératrices pour l'ordonnancement de tâches apériodiques dans les systèmes temps-réel distribués à contraintes strictes*, LISI Research report (Univ. Poitiers, 2000).

[10] KNUTH D.E., *The art of computer programming, Vol. 1* (Addison-Wesley, 1973).

[11] SCHÜTZENBERGER M.P.,*On the definition of a family of automata*, Inform. and Contr. **4** 245-270 (1961).

[12] SCHÜTZENBERGER M.P., *On a theroem of R. Jungen* Proc. Amer. Soc. **13** 885-890 (1962).

[13] STANLEY R.P., *Enumerative combinatorics, Vol. 2* (Cambridge, 1999).